Міністерство освіти і науки України

Національний університет «Львівська політехніка»



ЗВІТ

Про виконання лабораторної роботи № 7

З дисципліни “Кросплатформні засоби програмування ”

Варіант-25

**Виконала:**

студ. групи КІ-35

Янчинська Н. Р.

**Прийняв:**

Іванов Ю. С.

Львів 2022

ПАРАМЕТРИЗОВАНЕ ПРОГРАМУВАННЯ

Мета: оволодіти навиками параметризованого програмування мовою Java.

ЗАВДАННЯ

1. Створити параметризований клас, що реалізує предметну область задану варіантом. Клас має містити мінімум 4 методи опрацювання даних включаючи розміщення та виймання елементів. Парні варіанти реалізують пошук мінімального елементу, непарні – максимального. Написати на мові Java та налагодити програму-драйвер для розробленого класу, яка мстить мінімум 2 різні класи екземпляри яких розмішуються у екземплярі розробленого класу-контейнеру. Програма має розміщуватися в пакеті

Група.Прізвище.Lab6 та володіти коментарями, які дозволять автоматично

згенерувати документацію до розробленого пакету.

2. Автоматично згенерувати документацію до розробленого пакету.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.

4. Дати відповідь на контрольні запитання.

ВАРІАНТИ ЗАВДАННЬ

25. Сховище товарів

Код програми:

Main.java

**package** KI35\_Yanchynska\_Lab7;

/\*\*

\* Class Main

\* **@version** 1.0

\*/

**public** **class** Main {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args)

{

**try**{

StorageOfGoods books = **new** StorageOfGoods();

books.put(**new** Book(100, 100, "Book1"));

books.put(**new** Book(150, 140, "Book2"));

books.put(**new** Book(220, 90, "Book3"));

System.***out***.println(books.get(1).getName());

System.***out***.println(books.getHeaviestItem().getName());

System.***out***.println(books.getWeight());

books.showAll();

StorageOfGoods pencils = **new** StorageOfGoods();

pencils.put(**new** Pencil(10, 8, "Pencil1"));

pencils.put(**new** Pencil(5, 4, "Pencil2"));

pencils.put(**new** Pencil(8, 9, "Pencil3"));

System.***out***.println(pencils.getHeaviestItem().getName());

System.***out***.println(pencils.getWeight());

pencils.showAll();

}

**catch** (Exception e)

{

System.***out***.println(e);

}

}

}

Book.java

**package** KI35\_Yanchynska\_Lab7;

/\*\*

\* Class Book

\* **@version** 1.0

\*/

**public** **class** Book **extends** Goods {

**int** weight, volume;

String name;

/\*\*

\* Constructor

\* **@param** weight

\* **@param** volume

\* **@param** name

\*/

**public** Book(**int** weight, **int** volume, String name)

{

**super**(weight, volume, name);

**this**.weight = weight;

**this**.volume = volume;

**this**.name = name;

}

}

Goods.java

**package** KI35\_Yanchynska\_Lab7;

/\*\*

\* Class Goods

\* **@version** 1.0

\*/

**public** **class** Goods {

**int** weight;

**int** volume;

String name;

/\*\*

\* Constructor

\* **@param** weight

\* **@param** volume

\* **@param** name

\*/

**public** Goods(**int** weight, **int** volume, String name)

{

**this**.weight = weight;

**this**.volume = volume;

**this**.name = name;

}

/\*\*

\* Method returns the weight of the item

\*/

**public** **int** getWeight() {**return** weight;}

/\*\*

\* Method returns the volume of the item

\*/

**public** **int** getVolume() {**return** volume;}

/\*\*

\* Method returns the name of the item

\*/

**public** String getName() {**return** name;}

/\*\*

\* Method prints info

\*/

**public** **void** print()

{

System.***out***.println("Name: " + getName());

System.***out***.println("Volume: " + getVolume());

System.***out***.println("Weight: " + getWeight());

}

}

Pencil.java

**package** KI35\_Yanchynska\_Lab7;

/\*\*

\* Class Pencil

\* **@version** 1.0

\*/

**public** **class** Pencil **extends** Goods {

**int** weight, volume;

String name;

/\*\*

\* Constructor

\* **@param** weight

\* **@param** volume

\* **@param** name

\*/

**public** Pencil(**int** weight, **int** volume, String name)

{

**super**(weight, volume, name);

**this**.weight = weight;

**this**.volume = volume;

**this**.name = name;

}

}

StorageOfGoods.java

**package** KI35\_Yanchynska\_Lab7;

**import** java.util.\*;

/\*\*

\* Class StorageOfGoods

\* **@version** 1.0

\*/

**public** **class** StorageOfGoods <T **extends** Goods> {

ArrayList<T> storage = **new** ArrayList<>();

/\*\*

\* Method returns an item by the index

\* **@param** index

\* **@throws** Exception

\*/

**public** T get(**int** index)

{

**try**{

**return** storage.get(index);

}

**catch** (Exception e)

{

**throw** e;

}

}

/\*\*

\* Method puts an item

\* **@param** item

\*/

**public** **void** put(T item)

{

**try** {

storage.add(item);

}

**catch** (Exception e)

{

**throw** e;

}

}

/\*\*

\* Method shows storage content

\*/

**public** **void** showAll()

{

**for**(**int** i = 0; i < storage.size(); i++)

{

**try**{

storage.get(i).print();

System.***out***.println("=====================");

}

**catch** (Exception e)

{

System.***out***.println(e);

}

}

}

/\*\*

\* Method shuffles the content

\*/

**public** **void** shake()

{

Collections.*shuffle*(storage);

}

/\*\*

\* Method returns total content weight

\*/

**public** **int** getWeight()

{

**int** result = 0;

**for**(**int** i = 0; i < storage.size(); i++)

{

result += storage.get(i).getWeight();

}

**return** result;

}

/\*\*

\* Method returns the heaviest item

\*/

**public** T getHeaviestItem()

{

**int** heaviestIdx = 0;

**for**(**int** i = 0; i < storage.size(); i++)

{

**if**(i == 0)

{

**continue**;

}

**if**(storage.get(i).getWeight() > storage.get(heaviestIdx).getWeight())

{

heaviestIdx = i;

}

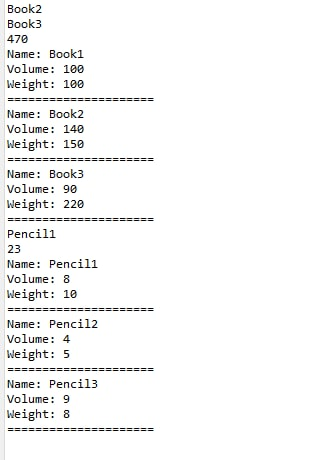
}

**return** storage.get(heaviestIdx);

}

}

Результат виконання програми:



Висновок:

На цій лабораторній роботі я ознайомилася та оволоділа навиками параметризованого програмування мовою Java.